Testing the Reliability of Inter-Rater Reliability

Brendan Eagan[†]
Educational Psychology
University of Wisconsin - Madison
Madison Wisconsin USA
beagan@wisc.edu

Jais Brohinsky
Educational Psychology
University of Wisconsin - Madison
Madison Wisconsin USA
brohinsky@wisc.edu

Jingyi Wang
Educational Psychology
University of Wisconsin - Madison
Madison Wisconsin USA
jwang862@wisc.edu

David Williamson Shaffer
Educational Psychology
University of Wisconsin - Madison
Madison Wisconsin USA
dws@education.wisc.edu

ABSTRACT

Analyses of learning often rely on coded data. One important aspect of coding is establishing reliability. Previous research has shown that the common approach for establishing coding reliability is seriously flawed in that it produces unacceptably high Type I error rates. This paper focuses on testing whether or not these error rates correspond to specific reliability metrics or a larger methodological problem. Our results show that the method for establishing reliability is not metric specific, and we suggest the adoption of new practices to control Type I error rates associated with establishing coding reliability.

CCS CONCEPTS

• General and reference ➞ Reliability • General and reference ➞ Empirical Studies • General and reference ➞ Measurement • General and reference ➞ Metrics • General and reference ➞ Validation

KEYWORDS

Interrater reliability, coding, reliability, validity, statistical analysis

ACM Reference format:

Brendan Eagan, Jais Brohinsky, Jingyi Wang, and David Williamson Shaffer. 2020. Testing the Reliability of Inter-Rater Reliability. In *Proceedings of the 10th International Conference on Learning Analytics & Knowledge (LAK'20), March 23-27, 2020, Frankfurt Germany. ACM, New York, NY, USA, 8 pages.* https://doi.org/10.1145/3375462.3375508

LAK '20, March 23–27, 2020, Frankfurt, Germany

ACM ISBN 978-1-4503-7712-6/20/03□$15.00
https://doi.org/10.1145/3375462.3375508

1 INTRODUCTION

In May of 2018, the *Journal of Learning Analytics* published a special issue exploring answers to the question, "What does methodology mean for learning analytics" [2]. In this issue, several articles discuss questions of reliability in models [3, 5, 14, 19], but there was no discussion of *inter-rater reliability* (IRR). This is problematic, as the reliability of any model depends on the reliability of the *inputs* to the model. In many analyses of learning, model inputs consist of coded data [8, 16, 20]. Thus, we argue, methodological questions about coding reliability are—and should be—important considerations for the field of learning analytics.

The general process for measuring IRR, or agreement between two coders, is to have each rater (human or machine) code a subset of the data, and then compute the rate of agreement using one of a number of possible measures. The measures most commonly used are the F statistic, Cohen's κ (hereafter, kappa), precision and recall, percent agreement, and percent positive agreement (also referred to as Jaccard's J). The value of the statistic computed is taken as a measure of agreement between the two raters.

In what follows, we examine the reliability of this process. We draw on research by Eagan and colleagues [10], which demonstrates that the standard method of establishing IRR introduces high Type I error rates with kappa, one of the most widely used IRR metrics in learning analytics. Briefly, they showed that finding agreement above a given kappa level between two raters (human and/or machine) on a subset of data did *not* provide a valid statistical warrant for concluding that the actual rate of agreement was above the desired value unless the subset was larger than those typically used in studies involving human coders. In other words, IRR measures computed on samples from larger datasets are in most cases inappropriately generalized.

Here, we extend this line of inquiry to other common IRR metrics, asking whether the problem uncovered by Eagan and colleagues is a more general problem with the method by which IRR is currently measured regardless of which statistic is used. We conclude that IRR involving human coders, as it is currently practiced in many studies in the field of learning analytics, is unreliable. However, by leveraging the conceptual and statistical problems we identify, we are able to construct a solution space for the problem. We then describe an alternative approach that uses a statistical control for Type I error in IRR measurement more broadly.

2 THEORY

2.1 Coding Data

All models are grounded in data that facilitate the translation from phenomena to interpretation. In many fields, *coding schemes* are used to organize data into categories [1, 12, 21]. These coded data can then be counted, compared, modeled, or otherwise analyzed to provide supporting or refuting evidence for some claim, or a justification for some action. In other words, coded data are crucial links in the chains of evidence substantiating the claims that emerge from a model. If a coding process doesn't identify what it purports to capture, conclusions or actions based on the model lose their claim to validity.

There are some approaches to coding and modeling that categorize data using a semantic or lexical model of a domain with no human input (e.g., topic modeling [6]). However, any *interpretation* of the meaning of those categories depends, at some point, on comparing the results with human judgement. Indeed, even in cases where raw data is fed directly into a model (e.g., neural networks [21]), the accuracy of the resulting model requires data that constitutes a *ground truth*. However, work in the social sciences often depends on placing a human "in the loop" at some point. Thus, questions of reliability, and therefore IRR, are an essential component of doing valid research in learning analytics.

In this sense, coded data are a critical foundation of a researcher's ability to surface patterns, build models, draw inferences, and decide on appropriate actions. However, *coded data are not the data themselves!* [1] For this reason, Hammer and Berland [15] suggest that codes are more aptly recognized as *claims* rather than *evidence*. As with all claims, there is uncertainty associated with coding, and IRR metrics are a means to quantify that uncertainty by measuring *agreement between two coding processes* using some particular metric for "agreement."

While there are approaches to coding that use ordinal or continuous scales, human raters are notoriously bad at calibrating ratings across coding instances [4], and many codes are more appropriately modeled using a binary (present/not present) decision [24]. In what follows we consider the case of IRR for binary coding schemes, although many of the same concerns apply to ordinal and continuous scales.[2]

2.2 IRR Metrics

There are number of IRR metrics, including percent agreement, Holsti's method, Scott's pi, Spearman's rho, Pearson's correlation coefficient, percent positive agreement (also known as Jaccard's J), Lin's concordance correlation coefficient, precision and recall, F statistic, the Kupper-Hafner index, or Krippendorff's alpha. With all of these methods, an *IRR score* is calculated based on a contingency table showing the number of times the raters agreed that the code was present or not present, and also the number of times that one thought it was present and the other did not.[3] (See Table 1 for the general structure of a contingency table for coding.) The processes for calculating five of the most commonly-used IRR metrics is shown in Table 2.

Table 1: Rater Agreement Contingency Table

		Second Rater	
		Thinks code is present	Thinks code is not present
First Rater	Thinks code is present	*Positive Agreement* (PP)	*Disagreement* (PN)
	Thinks code is not present	*Disagreement* (NP)	*Negative Agreement* (NN)

Each of these measures is sensitive to different properties of the data, such as the *base rate* of the code in the data (the frequency with which it occurs) and the number of pieces of data both raters coded.

Table 2: Common IRR measures

IRR measure	Definition	Equation
Precision (PR)	Measures the likelihood that the first rater thinks the code is present if the second rater thinks the code is present.	$\mathrm{PR} = \frac{\mathrm{PP}}{\mathrm{PP} + \mathrm{NP}}$
Recall (RC)	Measures the likelihood that the second rater thinks the code is present if the first rater thinks the code is present.	$\mathrm{RC} = \frac{\mathrm{PP}}{\mathrm{PP} + \mathrm{PN}}$
F Statistic (F)	Measures the harmonic mean of two raters' precision and recall.	$\mathrm{F} = 2(\frac{\mathrm{PR\ x\ RC}}{\mathrm{PR} + \mathrm{RC}})$

[1] There is a broader discussion of the relationship between *features* in data and the *selection* of those features on one hand, and the validity of inferences drawn from models based on those features on the other. Here, we are considering only the particular—but prevalent and important—case where human judgements are used to create some form of *gold standard* or *ground truth* in coding data.

[2] For a lengthier discussion of the import of binary coding see Shaffer [23].

[3] IRR uses agreement in the application of a code as a proxy for agreement in the concept of a code. This is, perhaps, most evident in automated coding where a computer cannot (yet) be said to *understand* a code, though it certainly can find it.

Jaccard's J (J) or Percent Positive Agreement	Measures the likelihood that both raters think the code is present if either rater thinks the code is present. Note: this is stricter than precision and recall because it accounts for all disagreement.	$\mathrm{J} = \frac{\mathrm{PP}}{\mathrm{PP} + \mathrm{PN} + \mathrm{NP}}$
Cohen's Kappa (κ)	Measures the ratio of two raters' observed agreement to perfect agreement, while controlling for chance.	$\kappa = \frac{\mathrm{OA} - \mathrm{PAC}}{1 - \mathrm{PAC}}$ $\mathrm{OA} = \frac{\mathrm{PP} + \mathrm{NN}}{\mathrm{PP} + \mathrm{PN} + \mathrm{NP} + \mathrm{NN}}$ $\mathrm{PAC} = (\mathrm{BR1} \text{ x } \mathrm{BR2}) + (1 - \mathrm{BR1})(1 - \mathrm{BR2})$ Where OA = Observed Agreement, PAC = Probability of Agreement, BR1 = base rate of the code for rater 1, and BR2 = base rate of the code for rater 2

2.3 Methods for Measuring IRR

Broadly speaking, there are two main approaches to measuring IRR using such statistics. The first is for two processes (usually two humans) to code all of the data. The measured rate of agreement in this approach is thus the true rate of agreement between the two processes.[4]

The second—and more common—approach to IRR uses a similar method. It, too, begins with two processes (hereafter, raters) coding the same data. However, in the second approach, the raters code only a subset of the data, often referred to as a test set.

Regardless of the IRR metric used, this second approach has been referred to as the Common Method for IRR Measurement (hereafter, the Common Method [10, 23]). The Common Method unfolds as follows (see also Figure 1):

1. The code is defined.
2. An IRR metric is chosen and a minimum value for acceptable agreement is set.
3. A test set of a specified size is randomly selected from the dataset.
4. Two raters independently code the test set.
5. The agreement of their coding is measured using the chosen IRR metric.
6. The IRR measure is compared to the minimum value in Step 2.
    a. If the IRR is **below** the minimum value, the raters resolve their disagreements, which can involve changing definition of the code, and repeat steps 3-5.
    b. If the IRR is **above** the minimum value, researchers coding is considered to be reliable.

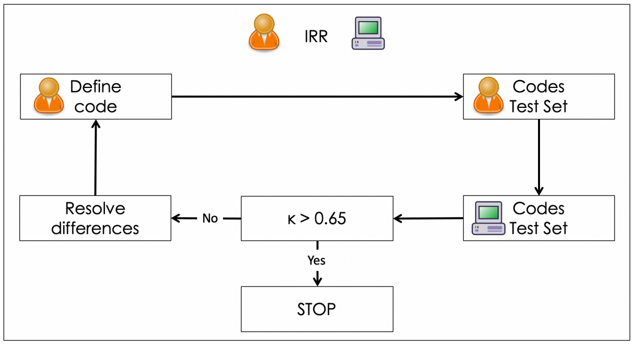

Figure 1: Workflow for establishing inter rater reliability using the Common Method (shown here with kappa).

2.4 Potential Errors in the Common Method

Although it is possible to achieve acceptable IRR in the first test set, it is more common to see raters coding multiple test sets before getting acceptable rates of agreement. This means that the actual number of excerpts coded by a human rater may be a significant rate-limiting factor in the Common Method.

More important, however, the Common Method relies on an implicit assumption that the IRR measured in the test set is equivalent to the true IRR that would be measured if both raters coded all of the data. For instance, if an IRR metric is reported at 0.90, it is assumed that if the two raters were to keep coding, their IRR would continue to be 0.90.

In other words, the Common Method is making a specific claim: the IRR metric from a sample (a test set) generalizes to a population (all of the data if coded by both raters). Any such generalization is potentially subject to *Type I errors*,[5] which occur when a false conclusion is made about a population based on the properties of a

[4] We are not claiming that the rate of agreement is "true" in any philosophical sense of the word, but only that the two processes have coded all the data and that we have quantified (using some measure) the rate of agreement. We should note, however, that in cases where two raters code all the data, it is mode common not to *report* the level of agreement, but rather to use *social moderation* [18] to reach a point of 100% agreement between the two coders. That is, the raters resolve their disagreements and come up with a single set of codes for the data.

[5] Type I errors are also known as *false positives*. The Type I error rate = (*false positives / all test sets with IRR measured above the minimum rate of acceptable agreement*). This is explained further in Table 4.

sample—in this case, if the IRR measured in a test set is *above* the minimum level of agreement, but the true rate of agreement that would be achieved if the two raters were to code the entire set is below the minimum level.

This raises two issues. First, it is not clear what an acceptable minimum rate for acceptable agreement should be. Kappa is often considered "reliable" at 0.65, but the other four most commonly used metrics have no agreed upon minimum.[6] The choice of an acceptable level of agreement thus depends on the standards of research domain in which the coding is used, what decisions or consequential inferences will be made based on analyses of the coded data, as well as factors like the potential repercussions associated with Type I and Type II errors.

Second, and perhaps more significant, the Common Method has no provision for estimating the rate of Type I errors. Without controlling for Type I errors, there is no statistically valid claim that the IRR established for a sample actually applies to the entire dataset from which the sample was drawn.

This raises a natural question: What is the impact of not controlling Type I errors, under the conditions raters usually encounter, in the field of learning analytics?

2.5 Monte Carlo Studies

Monte Carlo (MC) studies are commonly used to investigate questions about the performance and reliability of statistical tests in educational and psychological research [17]. MC studies are based on replication: a large number of simulated datasets (*replicates*) are generated, and a test statistic is calculated for each replicate. The number of replicates is determined by the repetitions needed to achieve statistical confidence in the result.

Critical to this process is the ability to construct simulated datasets that reflect the properties of the phenomenon in question. In the case of IRR, MC studies require construction of a *simulated codeset* representing a complete dataset as coded by two raters. Mathematically, this is represented by a set of binary ordered pairs — (1,1); (1,0); (0,1); or (0,0)—where the first number represents whether the code was applied by the first rater and the second number represents whether the code was applied by the second rater. (These correspond to the PP, PN, NP, NN combinations in Table 1).

Eagan and colleagues [10] constructed such simulated codesets by generating random pairs of 1s and 0s with a specified frequency to represent the codes of the first rater, and then permuted the first rater's codes to achieve specified parameters varied at random. (This process is further explained in the methods below). They used MC studies to demonstrate that kappa had high (greater than $\alpha = 0.05$) Type I error rates when IRR is calculated using the Common Method under typical conditions.

2.6 Research questions

In what follows, we adopt this method to assess the performance of five IRR measures commonly used in learning analytics (see Table 2), including kappa to check for replicability between our MC simulations and previous work. Eagan et al. [10] also demonstrated that the error rate of kappa is sensitive to the parameters of base rate and test set size. We therefore conducted MC simulations at multiple test set sizes and base rates.

Because there are no established standards for acceptable minimum rates of agreement, choosing an appropriate minimum level for an IRR statistic depends on the statistic chosen and the context in which the coding is used [11, 23]—and, in any event, establishing such levels is beyond the scope of the current paper.[7] As a result, we chose to explore the problems associated with the Common Method at three different minimum rates of agreement, from the lower end of those used in empirical studies to the higher end of rates seen in the literature.

We conduct this set of MC studies across five IRR measures to address the following research questions:

RQ1: Are the high Type I error rates associated with the Common Method under typical coding conditions involving a human rater specific to kappa, or do they pertain to the F statistic as well?

RQ2: Do the most commonly used IRR measures have different Type I error rates under typical coding conditions involving a human rater?

3 METHODS

3.1 Generation of Simulated Codesets

We identified four parameters that would uniquely define a simulated codeset: base rate of the code, codeset length (number of items to be coded), a target kappa value, and a target precision value.

For each simulated codeset, we used base rate and length to produce a unique set of codes at perfect agreement.[8] That is, we constructed a set of ordered pairs representing the codes for each piece of data in the simulated code set as a series of (1,1) followed by a series of (0,0) where the total number of ordered pairs (1,1) was equal to *base rate × length of the simulated codeset*.

We then used the target kappa value to change a subset of the ordered pairs (1,1) to (1,0) and a subset of the ordered pairs (0,0) to (0,1). That is, we introduced error in the coding so as to produce the target kappa level. Because kappa does not distinguish between positive and negative agreements, we used the target precision value to determine the proportion of (1,1)→(1,0) changes into (0,0)→(0,1) changes.

[6] Kappa is sometimes considered "reliable" at 0.65, but Cohen [7] provided no justification for this choice, and agreement at that rate often provides miscoded data at a rate high enough to jeopardize face validity for coding.

[7] An empirical approach to this issue is discussed in Eagan et al. [11].

[8] Because the IRR metrics we tested were invariant to permutation, we did not need to consider the order of the ordered pairs in the codeset.

A meta-analysis by Eagan and colleagues' [10] found limited guidance in the literature regarding appropriate parameter ranges of base rate, kappa, and precision during the coding process. This was due to the fact that most studies report only a final kappa value and do not provide base rates, test set length, or other information about the coding process.

Therefore, for our MC simulations, we empirically derived conservative estimates of what two trained human raters would reasonably produce for base rate, kappa and precision (see Table 3), based on the performance of raters observed in our own lab. Nearly 75% of the discourse codes used in our lab have base rates below 0.10. We believe our chosen parameters are not atypical in the kinds of data used by learning analytics researchers.

Table 3: Simulated Data Generation and MC Parameters and Ranges

Simulated Data Generation Parameters	**Parameter Ranges**
Base Rate	0.05, 0.10
Simulated Codeset Length	10,000
Kappa	0.30 – 1.00
Precision	0.60 – 1.00
MC Parameters	**Parameter Ranges**
Test set size	20, 40, 80, 200, 400, 800, 2000, 4000, 8000
Number of replicates	12,000

3.2 MC simulation construction

Using the codeset generation method described above, we employed the simulated IRR measurement (SIM) method [10] to model the Common Method based on three additional parameters: test set size, number of replicates, and minimum rate for acceptable agreement.

We chose test set sizes representing a range of values (a) lower than would be typically used by human coders (20, 40); (b) from the range of values typically used by human coders (80, 200, 400); and (c) larger than would be typically used by human coders but are sometimes used in machine learning applications (800, 2000, 4000, 8000). We chose a number of replicates to determine Type I error rates by incrementally increasing the number of replicates until the standard deviation of the Type I error rates decreased to less than or equal to 0.01. We found that 12,000 replicates were sufficient given the other parameters in our MC studies.

To complete each MC study for all five IRR metrics, we applied the SIM method as follows, using parameter values from Table 3:

1. We chose a base rate and test set size and created 12,000 simulated codesets using the generation method described above.
2. We computed the IRR metric for each simulated codeset, which represented the true rate of agreement that would be achieved if two raters had coded the entire dataset.
3. From each of these simulated codesets, we randomly selected a test at a given test set size (Common Method Step 3). This represented the number of excerpts raters would code in establishing IRR (Common Method Step 4).
4. We computed the IRR metric on each test set (Common Method Step 5).

For each study, this resulted in 24,000 numbers (two for each replicate): 12,000 true IRR values (one for each replicate), and 12,000 IRR values computed on one test set from each replicate. We produced a contingency table, as shown in Table 4, and computed the *Type I error rate = T1/(PP+T1)*.

Table 4: Type I Error Contingency Table

		Test set IRR	
		Above minimum rate	Not above minimum rate
True IRR	Above minimum rate	*Positive Agreement* (PP)	*Type II error (T2)*
	Not above minimum rate	*Type I error (T1)*	*Negative Agreement* (NN)

For RQ1, we selected the F statistic because it is, along with kappa, one of the most commonly used IRR metrics in the learning analytics field. Because the F statistic does not have a standardized minimum value of acceptable agreement, we chose to test its performance at three levels that span a typically reported range (0.50, 0.70, 0.90).

For RQ2, we repeated this MC process for each IRR metric using all combinations of chosen base rates and test set lengths at the median minimum value of acceptable agreement (0.70) from RQ1.

4 RESULTS

RQ1: Are the high Type I error rates associated with the Common Method under typical coding conditions involving a human rater specific to kappa, or do they pertain to the F statistic as well?

Table 4 shows the Type I error rates of the F Statistic for codes with base rates of 0.05 at all combinations of test set size and minimum rate of acceptable agreement that we considered. Of the 27 simulations we conducted, 18, or two thirds, had Type I error rates greater than 0.05. Of these 18, 10 had Type I error rates greater than 0.20.

Table 5 shows the Type I error rates of the F Statistic for codes with base rates of 0.10 at all combinations of test set size and minimum rate of acceptable agreement that we considered. Of the 27 simulations we conducted, 15, or just over half, had Type I error rates greater than 0.05. Of these 15, 7 had Type I error rates greater than 0.20.

We conclude from these MC studies that under many realistic conditions under which IRR is computed, the Common Method produces high Type I error rates. Regardless of base rate, the Common Method does not perform well unless the minimum required rate of agreement is high (F > 0.09). This is consistent with previous results found for MC studies of kappa.

Table 4: SIM method using F Statistic Type I error rates – for codes with base rate 0.05.

		Test Set Size								
		20	40	80	200	400	800	2000	4000	8000
F Statistic minimum rate of acceptable agreement	0.5	0.695	0.547	0.382	0.271	0.218	0.146	0.099	0.056	0.022*
	0.7	0.478	0.330	0.229	0.164	0.099	0.066	0.036*	0.026*	0.010*
	0.9	0.489	0.304	0.139	0.067	0.041*	0.027*	0.014*	0.010*	0.004*

* indicates Type I error rate less than 0.05

Table 5: SIM method using F Statistic Type I error rates – for codes with base rate 0.10.

		Test Set Size								
		20	40	80	200	400	800	2000	4000	8000
F Statistic minimum rate of acceptable agreement	0.5	0.545	0.387	0.303	0.230	0.174	0.126	0.074	0.041*	0.018*
	0.7	0.331	0.236	0.192	0.111	0.075	0.049*	0.033*	0.017*	0.008*
	0.9	0.303	0.153	0.085	0.046*	0.028*	0.020*	0.012*	0.007*	0.003*

* indicates Type I error rate less than 0.05

Table 6: SIM method using Precision, Recall, Jaccard's J, and Kappa (BR 0.05, minimum acceptable agreement 0.7)

		Test Set Size								
		20	40	80	200	400	800	2000	4000	8000
0.7 minimum rate of acceptable agreement	Precision	0.570	0.524	0.445	0.408	0.354	0.286	0.212	0.125	0.0543
	Recall	0.500	0.310	0.182	0.097	0.066	0.046*	0.026*	0.015*	0.006*
	Jaccard's J	0.460	0.288	0.174	0.090	0.064	0.037*	0.024*	0.014*	0.007*
	Kappa	0.477	0.338	0.239	0.154	0.095	0.061	0.042*	0.025*	0.009*

* indicates Type I error rate less than 0.05

RQ2: Do the most commonly used IRR measures have different Type I error rates under typical conditions of coding involving a human rater?

After conducting MC studies for the F Statistic with each combination of base rate, test set size, and minimum rate of agreement reported above, we ran simulations for each of the other IRR metrics of interest (Precision, Recall, Jaccard's J, and kappa). For these MC studies, we maintained the same range of test set sizes, but chose the common base rate of 0.05 and the median rate of minimum acceptable agreement (0.70).

Table 6 shows the Type I error rates for Precision, Recall, Jaccard's J, and kappa for codes with base rate 0.05, a minimum rate of acceptable agreement of 0.70, and all combinations of test set size. We can see that of the 36 simulations we conducted, 25 had Type I error rates greater than 0.05. Of these 25, 14 had Type I error rates greater than 0.20. Acceptable Type I error rates were only achieved in test set sizes of 800 or larger. In addition, in the ranges we examined, Precision never had acceptable Type I error rates.

We thus conclude that while Type I error rates do vary between different IRR statistics, no statistic performs well across the majority of the range of conditions typically found in studies involving human raters.

5 DISCUSSION

Previous work [10] has shown that the Common Method for establishing IRR introduces high Type I error rates for kappa. The results of our MC studies here suggest that the Common Method introduces unacceptable Type I error rates not just for kappa, but for other frequently used metrics at combinations of parameters typically used in the learning analytics community. This finding introduces concerns about the reliability of research claims based on coded data produced by the Common Method and contributes to the broader investigation of the role of reliability in learning analytics methodologies.

More specifically, our MC studies indicate that, under conditions typical in studies involving human coders, Type I error rates begin to fall below 0.05 as test set size and minimum acceptable rate of agreement increase. However, the test set sizes at which this result is achieved are beyond the capacity of most human raters, especially considering that most analyses rely on multiple codes and multiple iterations of testing for each code. Using a test set of length 400 might involve coding 1500-2000 pieces of data *for each code in the analysis*. Thus, test sets large enough to ensure low Type I error rates may be unfeasible using the Common Method.

While all five IRR metrics exhibit the same Type I error rates when used with the Common Method, Precision performed particularly poorly. Even when using test set sizes of 8,000, the Common

Method using Precision fails to achieve acceptable Type I error rates for a low base rate code at a typical minimum acceptable rate of agreement (Precision > 0.70). It is possible that Precision performs poorly in these MC studies because we used Precision as a parameter to generate the simulated codesets. However, in other work we have used Recall as a parameter for the generation of simulated codesets, and the same problem persists with Precision, which is particularly sensitive to coding errors in low base rate codes. Because there are few instances of the code, errors that either remove or add positive examples have dramatic effects on Precision.

These results highlight a number of conceptual and statistical problems associated with the Common Method. First, whenever IRR is calculated on a subset of data following the Common Method, there is an inherent issue of generalization, regardless of the IRR metric used. Second, problems with the Common Method persist even at relatively high criteria for acceptable agreement. Lastly, our results also identify broader statistical problems involving Type I error rates associated with the Common Method. For instance, the lower the base rate of a code, the more severe the Type I error rates. Similarly, the lower the minimum level of agreement, the more severe the Type I error rates.

These issues with the Common Method provide the outline for the issues that need to be resolved in order to address shortcomings in current IRR practices. The results of our study suggest that a viable solution must:

1. Work across different IRR metrics,
2. Be applicable beyond the observed sample agreement (i.e., have acceptable Type I error rates)
3. Perform well for low base rate codes,
4. Be compatible with a method for determining appropriate, and therefore variable, minimum levels of acceptable agreement.

The requirement for a solution to work across different IRR metrics is indicative of the unreliability of the Common Method itself. The foundational nature of this problem suggests that what is needed is not a new *statistic*, but rather a method that works in conjunction with existing statistics by measuring and controlling for Type I errors and thus providing valid warrants for generalizing from a sample of data coded by two raters to their expected rate of agreement across a larger dataset.

Because of the prevalence of important codes that may occur infrequently in learning analytics data, a successful solution will ideally perform well for low base rate codes.

And finally, given the lack of well-justified rates of agreement for most IRR statistics, an idea solution will also make it possible to determine appropriate minimum levels of acceptable agreement given the specific statistical claim being made. That is, researchers need to be able to establish that coding is *reliable enough* for some specific analysis, task, or decision [11].

This study has several limitations. First, we only investigated five IRR metrics. There are many others, although they are not frequently used. More importantly, we have no reason to believe the Common Method would perform better with any of them. Second, our study does not focus on the use of IRR between two machine raters. In those cases, IRR can be established with test sets that exceed the ranges we considered. However, even in these circumstances, large amounts of human-coded data are often used to establish validity and reliability of one, if not both, of the machine raters, thereby potentially introducing the Type I error rates documented above. Finally, not all learning analytics research uses IRR. While IRR employed through the Common Method is problematic, we do advocate using some approach (e.g., rho) to establish warrants for the claims comprising learning analytics research. These warrants ensure that results from the field are reliable as they are recommended to the educational designers, instructors, and students.

The unreliability of the Common Method has important consequences for IRR, and thus for any research involving human coders using binary codes. It means that humans either need to code far more data than has been used in many prior studies, or adjustments need to be made to the Common Method to control for Type I errors.

It is beyond the scope of this paper to explore this issue in detail, but we note that *Shaffer's rho* [9, 10, 23] is one approach to control for Type I error when using IRR metrics. Rho is a Monte Carlo rejective method that addresses all four of the criteria outlined above.

Briefly, rho is a method for controlling for Type I error in IRR statistics that do not have known distributions (which includes all metrics that we know of for binary coding; see Shaffer [23] for more details on rho).

Rho is a Monte Carlo rejective method that creates a large number of simulated data sets that conform to the null hypothesis: in this case, a large number of data sets with properties of the original data (e.g., code frequency) that have *agreement below the chosen threshold*. Rho then uses whatever sampling procedure was used to generate the original sample (that is, either random or conditional sampling) to take a sample of each data set under the null hypothesis. For each of these Monte Carlo samples, the value of the IRR statistic being used is computed. This produces an *empirical distribution of the IRR statistic under the null hypothesis with the given conditions of data and sampling procedure*. The rho statistic represents the percentage of samples in the empirical distribution of the IRR statistic that are more extreme than the IRR value observed in the actual sample. Thus, rho performs a similar function to a t-test in providing a bound on the expected Type I error rate in generalizing from a sample to a population.

As a result of the way rho is computed, it meets the criteria for addressing problems with the Common Methods. Specifically, rho (1) is independent of statistic used, and (2) controls for Type I error. Moreover, because rho is an empirical rejective method, it is accurate when conditional sampling is used. Thus, (3) rho can warrant generalizations in situations where positive instances of a code are oversampled, improving the efficiency of IRR measures for low frequency codes—that is, reducing the amount of data human raters need to code. Finally, in contrast to analytic distribution-based approaches (e.g. FCE [13]), rho can be used with any minimum level of acceptable agreement. As a result, (4) it is possible to include rho in Monte Carlo methods to estimate the level of agreement required for a statistical result to remain valid (see Eagan et al. [11] for more details.)

Whether researchers use rho or some other technique for controlling the Type I errors associated with establishing IRR, our results indicate that the reliability problems associated with the Common Method persist across standard IRR metrics in situations researchers are likely to encounter. These issues are fundamental to any analytic claims relying on IRR in the evidentiary chain from data to meaning. Moreover, these concerns offer a unique opportunity for integrating solutions into the emerging learning analytics community as it coheres and establishes its methodological boundaries.

ACKNOWLEDGMENTS

This work was funded in part by the National Science Foundation (DRL-1661036, DRL-1713110), the Wisconsin Alumni Research Foundation, and the Office of the Vice Chancellor for Research and Graduate Education at the University of Wisconsin-Madison. The opinions, findings, and conclusions do not reflect the views of the funding agencies, cooperating institutions, or other individuals.

REFERENCES

[1] Ron Artstein & Massimo Poesio. 2008. Inter-coder agreement for computational linguistics. *Computational Linguistics* 34, 4, 555-596.

[2] Yoav Bergner, Geraldine Gray, & Charles Lang. 2018. What does methodology mean for learning analytics? *Journal of Learning Analytics*, 5, 2, 1-8.

[3] Nigel Bosch & Luc Paquette. 2018. Metrics for discrete student models: chance levels, comparisons, and use cases. *Journal of Learning Analytics*, 5, 2, 86-104.

[4] Robert L. Brennan. 2001. *Generalizability theory.* Springer-Verlag.

[5] Matthieu J. S. Brinkhuis, Alexander O. Savi, Abe D. Hofman, Frederik Coomans, Han L. J. van Der Maas, & Gunter Maris. 2018. Learning as it happens: a decade of analyzing and shaping a large-scale online learning system. *Journal of Learning Analytics,* 5, 2, 29-46.

[6] Jonathan Chang, Sean Gerrish, Chong Wang, Jordan L. Boyd-Graber, & David M. Blei. 2009. Reading tea leaves: How humans interpret topic models. In *Advances in neural information processing system – Proceedings of the 2009 Conference*, 288-296.

[7] Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20, 1, 37–46.

[8] Mollie Dollinger, Danny Liu, Natasha Arthars, & Jason M. Lodge. 2019. Working together in learning analytics towards the co-creation of value. *Journal of Learning Analytics,* 6, 2, 10-26.

[9] Brendan Eagan, Brad Rogers, Rebecca Pozen, Cody Marquart, & David W. Shaffer. 2016. rhoR: Rho for inter rater reliability (Version 1.1.0). Retrieved from https://cran.r-project.org/web/packages/rhoR/index.html

[10] Brendan Eagan, Bradley Rogers, Ronald Serlin, Andrew R. Ruis, Golnaz Arastoopour Irgens, David W. Shaffer. 2017. Can We Rely on Reliability? Testing the Assumptions of Inter-Rater Reliability. *Making a Difference: Prioritizing Equity and Access in CSCL: 12th International Conference on Computer-Supported Collaborative Learning*, eds. B. K. Smith, M. Borge, E. Mercier, & K. Y. Lim (2017), II:529–532.

[11] Brendan Eagan, Zachari Swiecki, Cayley Farrell, & David W. Shaffer. 2019. The binary replicate test: Determining the sensitivity of CSCL models to coding error. *Proceedings of the 13th International Conference on Computer-Supported Collaborative Learning*.

[12] George Forman. 2003. An extensive empirical study of feature selection metrics for text classification. *Journal of machine learning research*, 3, 1289-1305.

[13] Joseph Fleiss, Jacob Cohen, & Brian Everitt. 1969. Large sample standard errors of kappa and weighted kappa. *Psychological bulletin,*72, 5, 323-327.

[14] Josh Gardner & Christopher Brooks. 2018. Evaluating predictive models of student success: closing the methodological gap. *Journal of Learning Analytics,* 5, 2, 105-125.

[15] David Hammer & Leema K. Berland. 2014. Confusing claims for data: A critique of common practices for presenting qualitative research on learning. *Journal of the Learning Sciences*, 23, 1, 37-46.

[16] Fatima Harrak, François Bouchet, & Vanda Luengo. 2019. From student questions to student profiles in a blended learning environment. *Journal of Learning Analytics,* 6, 1, 54-84.

[17] Michael R. Harwell. 1992. Summarizing monte carlo results in methodological research. *Journal of Educational Statistics*, 17, 4, 297-313.

[18] Leslie R. Herrenkohl & Lindsay Cornelius. 2013. Investigating elementary students' scientific and historical argumentation. *Journal of the Learning Sciences*, 22, 3, 413-461.

[19] Benjamin A. Motz, Paulo F. Carvalho, Joshua R. de Leeuw, & Robert L. Goldstone. 2018. Embedding experiments: staking causal inference in authentic educational contexts. *Journal of Learning Analytics*, 5, 2, 47-59.

[20] Jun Oshima, Ritsuko Oshima, & Wataru Fujita. 2018. A mixed-methods approach to analyze shared epistemic agency in jigsaw instruction at multiple scales of temporality. *Journal of Learning Analytics*, 5, 1, 10-24.

[21] Zacharoula Papamitsiou & Anastasios A. Economides. 2014. Learning analytics and educational data mining in practice: A systematic literature review of empirical evidence. *Journal of Educational Technology & Society, 17*, 4, 49-64.

[22] Liam Rourke, Terry D. Anderson, D. Randy Garrison, & Walter Archer. 2001. Methodological issues in the content analysis of computer conference transcripts. *International Journal of Artificial Intelligence in Education*, 12, 8-22.

[23] David W. Shaffer. 2017. *Quantitative ethnography*. Madison, Wisconsin: Cathcart Press.

[24] Anthony J. Viera & Joanne M. Garrett. 2005. Understanding interobserver agreement: The kappa statistic. *Family Medicine*, 37, 5, 360-363.